Cancer Integration via Multikernel Learning (*CIMLR*)

***Daniele Ramazzotti*[1], *Avantika Lal*[1], *Bo Wang*[2], *Luca De Sano*[3], *Serafim Batzoglou*[2], *and Arend Sidow*[1]**

[1]Department of Pathology, Stanford University, CA 94305, USA.
[2]Department of Computer Science, Stanford University, CA 94305, USA.
[3]Dipartimento di Informatica Sistemistica e Comunicazione, Università degli Studi di Milano-Bicocca, Milano, Italy.

October 15, 2018

Overview. In this package we provide the implementation of the CIMLR method. This method was originally applied to cancer genomic data, but it is in principle capable of effectively and efficiently learning similarities in all the contexts where diverse and heterogeneous statistical characteristics of the data make the problem harder for standard approaches.

In this vignette, we give an overview of the package by presenting some of its main functions.

Contents

1 Changelog

1.0.0 Package release.

2 Algorithms and useful links

Acronym	Extended name	Reference
CIMLR	Cancer Integration via Multikernel Learning	Publication

3 Using the CIMLR R package

We first load the data given as examples in the package. The dataset named GliomasReduced is a reduced dataset of the data originally published in Cancer Genome Atlas Research Network. "Comprehensive, integrative genomic analysis of diffuse lower-grade gliomas." New England Journal of Medicine 372.26 (2015): 2481-2498, where multi-omic data for a cohort of patients affected by lower grade gliomas are provided.

```
library(CIMLR)
data(GliomasReduced)
```

Now, we first estimate the best number of clusters from data with two heuristics (we refer to the CIMLR paper for details). We note that these data are highly reduced and the results are not indicative of the full dataset.

```
set.seed(11111)
NUMC = 2:10
res_example = CIMLR_Estimate_Number_of_Clusters(GliomasReduced$in_X,
                                                NUMC = NUMC,
                                                cores.ratio = 0)
```

Best number of clusters, K1 heuristic:

```
NUMC[which.min(res_example$K1)]
```

```
## [1] 5
```

K2 heuristic:

```
NUMC[which.min(res_example$K2)]
```

```
## [1] 5
```

Results of the two heuristics:

```
res_example
```

```
## $K1
## [1]  -13.91113    5.56971   72.82487 -299.09427  238.02414  -13.12943  -15.93506
## [8]   18.22102  -17.71183
##
## $K2
## [1]  -20.86669    7.42628   91.03108 -358.91312  277.69483  -15.00506  -17.92695
## [8]   20.24557  -19.48301
```

We now run CIMLR an the above mentioned input data looking for a total of 5 subtypes, i.e., clusters.

```
set.seed(11111)
example = CIMLR(X = GliomasReduced$in_X, c = 5, cores.ratio = 0)
```

```
## Computing the multiple Kernels.
## Performing network diffusion.
## Iteration:  1
## Iteration:  2
## Iteration:  3
## Iteration:  4
## Iteration:  5
## Iteration:  6
## Iteration:  7
## Iteration:  8
## Iteration:  9
## Iteration:  10
## Performing t-SNE.
## Epoch: Iteration # 100  error is:  0.09743819
## Epoch: Iteration # 200  error is:  0.03390648
## Epoch: Iteration # 300  error is:  0.02610777
## Epoch: Iteration # 400  error is:  0.02562452
## Epoch: Iteration # 500  error is:  0.02517294
## Epoch: Iteration # 600  error is:  0.02476866
## Epoch: Iteration # 700  error is:  0.02442203
## Epoch: Iteration # 800  error is:  0.02410315
## Epoch: Iteration # 900  error is:  0.02381543
## Epoch: Iteration # 1000  error is:  0.02357839
## Performing Kmeans.
## Performing t-SNE.
## Epoch: Iteration # 100  error is:  20.62822
```

```
## Epoch: Iteration # 200  error is:  1.508242
## Epoch: Iteration # 300  error is:  0.9807764
## Epoch: Iteration # 400  error is:  0.6275994
## Epoch: Iteration # 500  error is:  0.5573287
## Epoch: Iteration # 600  error is:  0.3522938
## Epoch: Iteration # 700  error is:  4.993374
## Epoch: Iteration # 800  error is:  1.660517
## Epoch: Iteration # 900  error is:  2.2503
## Epoch: Iteration # 1000  error is:  3.825846
```

As a further understanding of the results, we now visualize the clusters in a plot to see how well they are separated.

```
plot(example$ydata,
    col = c(topo.colors(5))[example$y[["cluster"]]],
    xlab = "CIMLR component 1",
    ylab = "CIMLR component 2",
    pch = 20,
    main="CIMILR 2D visualization for GliomasReduced")
```

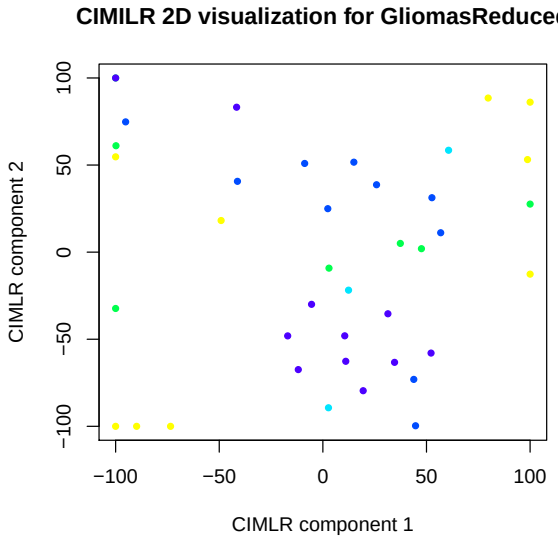

Figure 1: Visualization of the 5 clusters retrieved by CIMLR on the dataset GliomasReduced

We also run CIMLR feature ranking on the same inputs to get a rank of the key genes with the related pvalues.

```
set.seed(11111)
input_data = rbind(GliomasReduced$in_X$point_mutations,GliomasReduced$in_X$copy_numbers,
                   GliomasReduced$in_X$methylations,GliomasReduced$in_X$expression_values)
ranks = CIMLR_Feature_Ranking(A=example$S,X=input_data)
```

The output of this function contains pvalues for the features ordered by significance and a vector named aggR that provides the position of the corresponding pvalue in the original data.

```
head(ranks$pval)
## [1] 4.234257e-47 5.045487e-46 1.568507e-41 6.238639e-40 7.661582e-37 2.272592e-32
head(ranks$aggR)
## [1] 110 379 162 300 283 127
```

In this case, feature 110 in input data is the one with best pvalue of 4.234257e-47.

4 sessionInfo()

- R version 3.5.1 (2018-07-02), `x86_64-apple-darwin15.6.0`
- Locale: `C/it_IT.UTF-8/it_IT.UTF-8/C/it_IT.UTF-8/it_IT.UTF-8`
- Running under: `macOS High Sierra 10.13.6`
- Matrix products: default
- BLAS: `/Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRblas.0.dylib`
- LAPACK: `/Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRlapack.dylib`
- Base packages: base, datasets, grDevices, graphics, methods, stats, utils
- Other packages: CIMLR 1.0.0, knitr 1.20
- Loaded via a namespace (and not attached): BiocManager 1.30.2, BiocStyle 2.9.6, Matrix 1.2-14, Rcpp 0.12.19, backports 1.1.2, compiler 3.5.1, digest 0.6.17, evaluate 0.11, grid 3.5.1, highr 0.7, htmltools 0.3.6, lattice 0.20-35, magrittr 1.5, parallel 3.5.1, rmarkdown 1.10, rprojroot 1.3-2, stringi 1.2.4, stringr 1.3.1, tools 3.5.1, yaml 2.2.0